Exercise work 3

Name: Pekka Lehtola

Do you need help/comments:

Koodi paikoittain kommentoimatta, koska aika loppui kesken (Firman projekti vei viikosta suuren osan ajasta…)

Schedule:

- Week 1: Core game mechanics working.
- Week 2: UI, UML diagrams, Flow chart
- Week 3: Random events, Inheritance, Polymorphism. Ui class separated to multiple classes
- Final: Updated graphics, UML diagrams and Flow chart final version.

Challenge goal:

Challenging (Grade 4-5)

Work done this week: 8h

What has been implemented / changed:

Ui class separated to multiple modules / classes.

Inheritance from Ui class to other Ui elements.

Random events as power ups that appear into the level.

Ui elements:

```
import pygame
from config import *
from ui import Ui


class Heart_ui(Ui):

    def __init__(self):
        super(Heart_ui, self).__init__("images/hearth_1.png")
```

Heart Ui otherwise unmodified.

```
import pygame
from config import *
import time
from ui import Ui

class Ammo_ui(Ui):
    def __init__(self, image):

        super(Ammo_ui, self).__init__(image)

        # Used for characters reload animation.
        self.reload_list = ["images/circle1.png", "images/circle2.png", "images/circle3.png", "images/circle4.png"]
        self.counter = 0
```

Modifications to show_ammo function.

```
# Shows every ammo that character has in ammo list
def show_ammo(self, character, screen):

    y = 10

    if character.name == "Cowboy":

        x = 60

        for ammo in character.ammo:

            if ammo.super == True:

                ammo.rect.x = x
                ammo.rect.y = 90
                screen.blit(ammo.surf, ammo.rect)
            else:
                ammo.rect.y = y
                ammo.rect.x = x
                screen.blit(ammo.surf, ammo.rect)

                x += 30
```

Ammo class:

main.py | ammo.py | heart_ui.py | ammo_ui.py | heart_power_up.py | character.py | ui.py | config.py | sup

```python
# File name: character
# Author: Pekka Lehtola
# Description: Ammo class for characters to shoot.

import pygame
from config import *

# Ammo is derived from pygame sprites
class Ammo(pygame.sprite.Sprite):

    def __init__(self, ammo, name):
        super(Ammo, self).__init__()
        self.name = name
        self.surf = pygame.image.load(ammo)     # Image
        self.surf.set_colorkey(GREEN_SCREEN)    # See through color
        self.rect = self.surf.get_rect()        # Hitbox

        self.shot = False

        self.super = False

    # Function for shot ammo
    def ammo_shot(self, shooter, enemy):

        if self.shot == True:

            # If the shooter is indian arrow travels to left
            # If arrow hits cowboy, cowboy is killed.
            if shooter.name == "Indian":
                self.rect.move_ip(-BULLET_VELOCITY, 0)

                if self.rect.colliderect(enemy.rect):
                    self.rect.x = -2200
                    self.kill()

                    if self.super:

                        for heart in range(3):

                            if len(enemy.health) == 0:
                                enemy.kill()

                            else:
                                enemy.health.pop(-1)
                    else:

                        if len(enemy.health) == 0:
                            enemy.kill()

                        else:
                            enemy.health.pop(-1)

            # Same as indian but bullet travels right
            if shooter.name == "Cowboy":
                self.rect.move_ip(BULLET_VELOCITY, 0)

                if self.rect.colliderect(enemy.rect):
                    self.rect.x = 2200
                    self.kill()

                    if self.super:

                        for heart in range(3):

                            if len(enemy.health) == 0:
                                enemy.kill()

                            else:
                                enemy.health.pop(-1)
                    else:

                        if len(enemy.health) == 0:
                            enemy.kill()

                        else:
                            enemy.health.pop(-1)
```

Ui class:

main.py | ammo.py | heart_ui.py | ammo_ui.py | heart_power_up.py | character.py | ui.py | config.py | sup

```python
# File name: ui
# Author: Pekka Lehtola
# Description: User interface class. Used for showing different elements in screen.

import pygame
from config import *
from time import sleep
#from main import main


class Ui(pygame.sprite.Sprite):

    def __init__(self, image):
        super(Ui, self).__init__()
        self.surf = pygame.image.load(image)
        self.surf.set_colorkey(GREEN_SCREEN)
        self.rect = self.surf.get_rect()

    ### WORK IN PROGRESS ###

    # In the start of the game shows count down
    def show_element(self, screen):


        self.rect.centerx = (SCREEN_SIZE_HOR / 2)
        self.rect.centery = (SCREEN_SIZE_VER / 2)
        screen.blit(self.surf, self.rect)
        pygame.display.flip()
        sleep(1.2)

    # Will be used to show the winner.
    #def game_end(self):

    def countdown(self, screen):

        self.surf = pygame.image.load("images/33.png")
        self.surf.set_colorkey(GREEN_SCREEN)
        self.show_element(screen)
        screen.blit(BG, (0, 0))

        self.surf = pygame.image.load("images/22.png")
        self.surf.set_colorkey(GREEN_SCREEN)
        self.show_element(screen)
        screen.blit(BG, (0, 0))

        self.surf = pygame.image.load("images/11.png")
        self.surf.set_colorkey(GREEN_SCREEN)
        self.show_element(screen)
        screen.blit(BG, (0, 0))

    def winner(self, cowboy, indian, screen):


        if len(indian.health) == 0:

            self.surf = pygame.image.load("images/cowboy_wins.png")
            self.surf.set_colorkey(GREEN_SCREEN)
            self.show_element(screen)
            sleep(1)
            return True

        if len(cowboy.health) == 0:

            self.surf = pygame.image.load("images/indian_wins.png")
            self.surf.set_colorkey(GREEN_SCREEN)
            self.show_element(screen)
            sleep(1)
            return True
```

Main 1:

main.py | ammo.py | heart_ui.py | ammo_ui.py | heart_power_up.py | character.py | ui.py | config.py | sup

```python
# File name: main
# Author: Pekka Lehtola
# Description: main file exercise work

import pygame
from config import *
from character import *
from ui import *
from time import perf_counter, sleep
from heart_ui import *
from ammo_ui import *
from heart_power_up import *
from super_ammo import Ammo_power_up
from random import randint

running = True



def main():

        # Initialize pygame
        pygame.init()

        # Sets up screen size
        screen = pygame.display.set_mode([SCREEN_SIZE_HOR, SCREEN_SIZE_VER])

        heart_timer = randint(7, 12)
        super_ammo_timer = randint(14, 20)
        global running

        # Creates indian and cowboy objects.
        # Also define graphics to them.
        indian = Character("images/indian_1.png", "Indian")
        cowboy = Character("images/cowboy_1.png", "Cowboy")

        # Adds indian and cowboy to sprite group.
        all_sprites = pygame.sprite.Group()
        all_sprites.add(indian)
        all_sprites.add(cowboy)

        # Indians initial location
        indian.rect.x = 1820
        indian.rect.y = 640

        # Cowboys initial location
        cowboy.rect.x = 100
        cowboy.rect.y = 640

        heart = Heart_ui()
        reload = Ammo_ui("images/circle1.png")

        heart_power_up_list = []
        super_ammo_power_up_list = []

        heart.set_up_hearts(character=cowboy)
        heart.set_up_hearts(character=indian)

        # Background update
        screen.blit(BG, (0, 0))

        # start countdown
        countdown = Ui("images/11.png")
        countdown.countdown(screen)

        winner = Ui("images/cowboy_wins.png")
```

Main 2:

```
        winner = Ui("images/cowboy_wins.png")

        while running:

            timer = perf_counter()


            # Detects events while game is running.
            for event in pygame.event.get():
                if event.type == pygame.QUIT:
                    running = False

            # Pygame detects what keys are pressed.
            pressed_keys = pygame.key.get_pressed()

            # The term used for rendering objects is blitting.
            # Surf : Surface, Rect : Rectangle.
            # Rectangle is used for collide detection
            # Surface can be image or color.

            for characters in all_sprites:
                screen.blit(characters.surf, characters.rect)

            # Ui elements
            heart.show_hearts(cowboy, screen)
            heart.show_hearts(indian, screen)

            reload.show_ammo(cowboy, screen)
            reload.show_ammo(indian, screen)

            reload.show_reload(indian, screen)
            reload.show_reload(cowboy, screen)


            # Reload function
            indian.reload(pressed_keys, timer)
            cowboy.reload(pressed_keys, timer)

            # Shooting function
            indian.shoot(pressed_keys, timer)
            cowboy.shoot(pressed_keys, timer)

            # Moving function
            indian.move(pressed_keys)
            cowboy.move(pressed_keys)

            # Bullet travel and hit detection
            for ammo in indian.shot_ammo:

                ammo.ammo_shot(indian, cowboy)
                screen.blit(ammo.surf, ammo.rect)

            for ammo in cowboy.shot_ammo:

                ammo.ammo_shot(cowboy, indian)
                screen.blit(ammo.surf, ammo.rect)

```

Main 3:

```
            # Power ups

            if timer > heart_timer:

                heart_timer = timer + randint(7, 12)
                heart_power_up = Heart_power_up()
                heart_power_up.add_power_up(heart_power_up_list)

            for hearts in heart_power_up_list:
                hearts.show_power_up(screen)
                hearts.heart_pickup(indian, cowboy, heart)

            if timer > super_ammo_timer:

                super_ammo_timer = timer + randint(14, 20)
                super_ammo = Ammo_power_up()
                super_ammo.add_power_up(super_ammo_power_up_list)

            for super_ammo in super_ammo_power_up_list:

                super_ammo.show_power_up(screen)
                super_ammo.super_pickup(indian, cowboy)




            # Update screen
            pygame.display.flip()

            # Background update
            screen.blit(BG, (0,0))

            # Starts game over when either character dies.

            if winner.winner(cowboy, indian, screen):
                main()

        pygame.quit()

main()
```

Heart power up class:

main.py | ammo.py | heart_ui.py | ammo_ui.py | heart_power_up.py | character.py | ui.py | config.py | sup

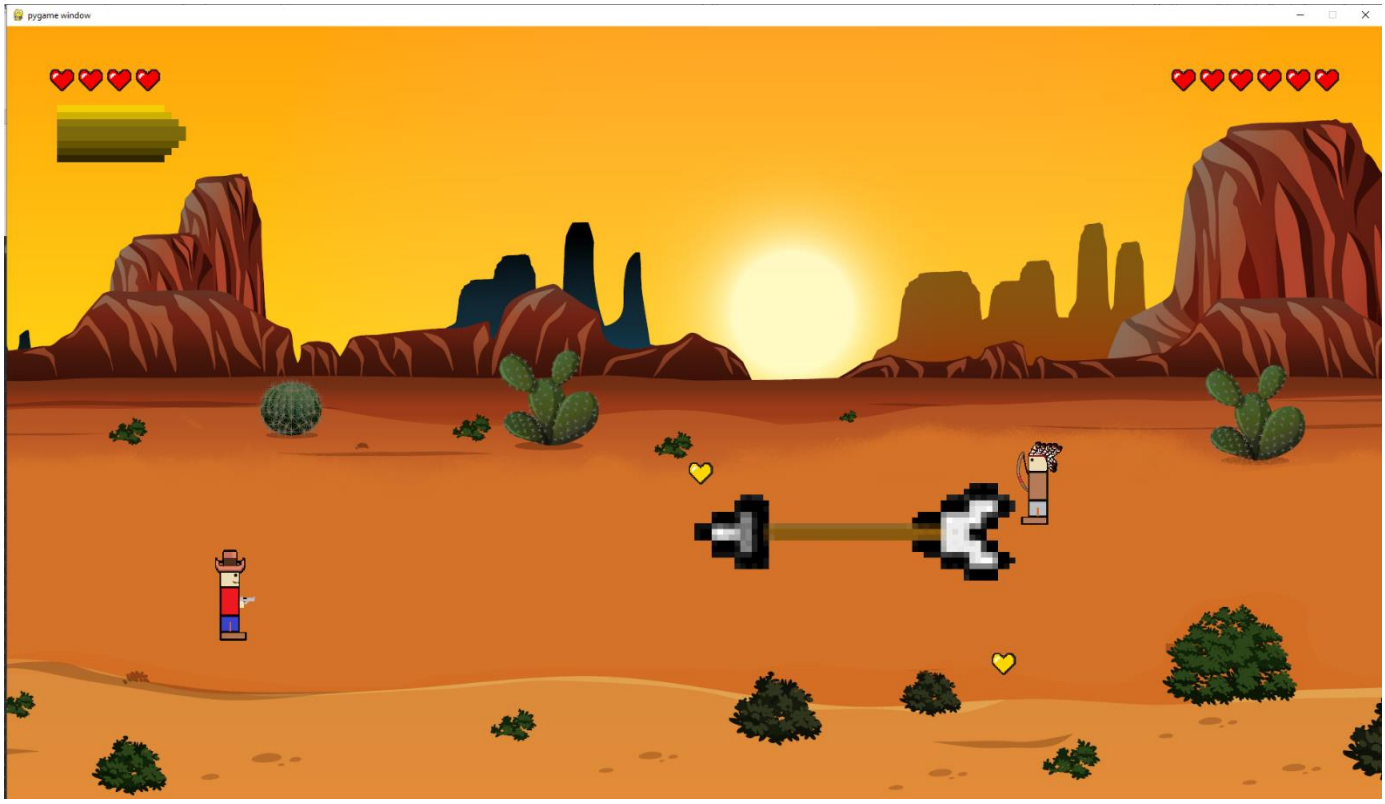
```python
# File name: hearth_power_up
# Author: Pekka Lehtola
# Description: Used for creating heart power ups in game.

from random import randint
from heart_ui import Heart_ui
from heart_ui import *

class Heart_power_up(Heart_ui):
    def __init__(self):
        super(Heart_power_up, self).__init__()
        self.surf = pygame.image.load("images/heart_gold.png")
        self.surf.set_colorkey(GREEN_SCREEN)


    def add_power_up(self, hearth_power_up_list):

        self.rect.centerx = randint(300, 1620)
        self.rect.centery = randint(510, 1000)

        hearth_power_up_list.append(self)

    def show_power_up(self, screen):

        screen.blit(self.surf, self.rect)

    def heart_pickup(self, indian, cowboy, heart):

        if self.rect.colliderect(cowboy.rect):
            cowboy.health.append(heart)
            self.rect.x = 2200
            self.kill()

        if self.rect.colliderect(indian.rect):
            indian.health.append(heart)
            self.rect.x = 2200
            self.kill()

```

Ammo power up class:

main.py | ammo.py | heart_ui.py | ammo_ui.py | heart_power_up.py | character.py | ui.py | config.py | sup

```python
# File name: super_ammo
# Author: Pekka Lehtola
# Description: super ammo used for ammo power ups.

from config import *
import pygame
from random import randint
from ammo import Ammo


class Ammo_power_up(pygame.sprite.Sprite):

    def __init__(self):
        super(Ammo_power_up, self).__init__()
        self.surf = pygame.image.load("images/super_ammo.png")
        self.surf.set_colorkey(GREEN_SCREEN)
        self.rect = self.surf.get_rect()



    def add_power_up(self, super_ammo_power_up_list):

        self.rect.centerx = randint(300, 1620)
        self.rect.centery = randint(510, 1000)

        super_ammo_power_up_list.append(self)

    def show_power_up(self, screen):

        screen.blit(self.surf, self.rect)

    def super_pickup(self, indian, cowboy):

        if self.rect.colliderect(cowboy.rect):

            self.rect.x = 2200
            self.kill()

            cowboy.ammo = []
            for ammo in range(0, 1):

                ammo = Ammo("images/bullet.png", "bullet")
                ammo.surf = pygame.image.load("images/bullet.png")
                ammo.surf = pygame.transform.scale(pygame.image.load("images/bullet.png"),
                                                   (bullet_x * 10, bullet_y * 10))
                ammo.surf.set_colorkey(GREEN_SCREEN)
                ammo.rect = ammo.surf.get_rect()
                ammo.super = True

                cowboy.ammo.append(ammo)

        if self.rect.colliderect(indian.rect):

            self.rect.x = 2200
            self.kill()

            indian.ammo = []

            for ammo in range(0, 1):

                ammo = Ammo("images/arrow.png", "arrow")
                ammo.surf = pygame.image.load("images/arrow.png")
                ammo.surf = pygame.transform.scale(pygame.image.load("images/arrow.png"),
                                                   (arrow_x * 8, arrow_y * 8))
                ammo.surf.set_colorkey(GREEN_SCREEN)
                ammo.rect = ammo.surf.get_rect()
                ammo.super = True

                indian.ammo.append(ammo)
```

Screen captures from the game:

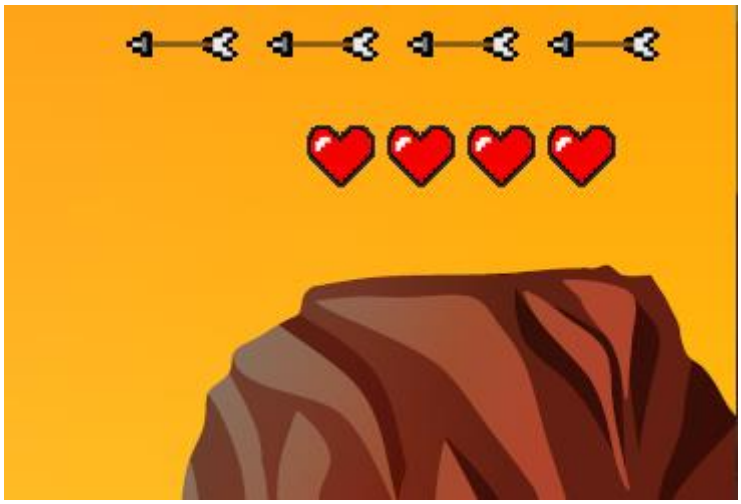


Kun pelaaja kerää super ammo power upin luodin koko ja vahinko minkä tuottaa kasvaa.

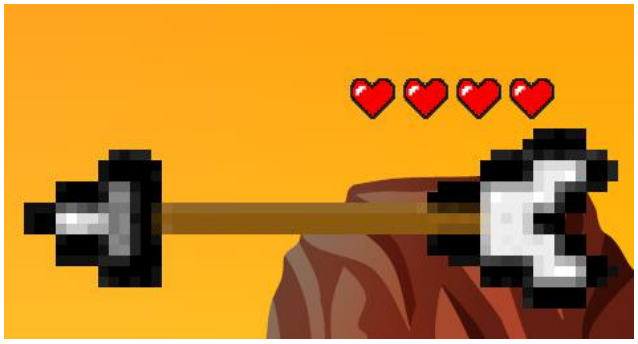
Kultaiset sydämmet antaa pelaajalle lisää elämää

Molemmat power upit ilmestyy kentälle satunnaisina aikoina

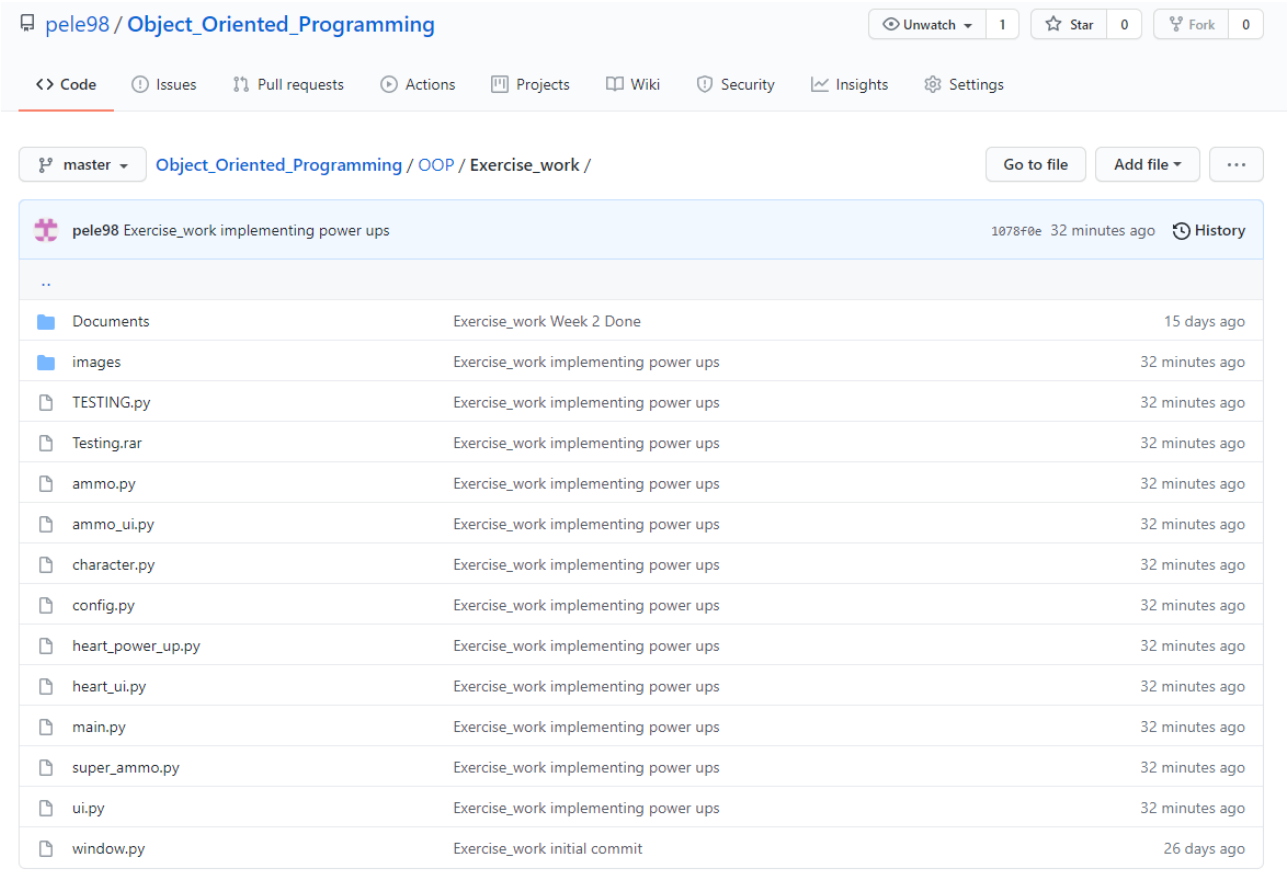
Normaalit ammukset:



Super ammukset:



Screen capture of git log (showing that you made a commit after every task).



Self-assessment:

This exercise was easy/difficult/ok/etc. for me because…

Ok. haasteena ilmeni poweruppien ilmestyminen satunnaisen ajan kuluttua ja tietylle alueelle.

Haaste oli myös saada ne toimimaan oikien.

Doing this exercise, I learned…

Pygame.scale metodin käyttöä

I am still wondering…

-

I understood/did not understand that… ; I did/did not know that… ; I did/did not manage to do…

Power upit toimivat viellä hiukan bugisesti ja tosiaan aika ei riittänyt koodin viimeistelyyn. Korjaan Asian viimeisessä palautuksessa.